

In the Claims

A copy of the pending claims is provided for the Examiner's reference:

1. (Previously presented) A method for connecting an emulated electronic device to a network operating at a specified bit-rate higher than the emulated electronic device, the method comprising:

providing a computer having a network interface, an emulation interface and a memory, the network interface being capable of communicating with the network at the specified bit-rate;

connecting the computer through the network interface with the network and through the emulation interface with the emulated electronic device; and

executing in the computer a network handling program which performs:

(a) receiving data packets from the network through a the network interface;

(b) storing the data packets received from the network in a first buffer in the memory;

(c) transmitting the data packets in the first buffer to the emulated electronic device through ~~an~~ the emulation interface;

(d) receiving data packets from the emulated electronic device through the emulation interface; and

(e) transmitting the data packets received from the emulated electronic

device to the network through the network interface.

2. (Previously presented) The method of Claim 1 further comprising storing the data packets received from the emulated electronic device in a second buffer in memory and wherein the step of transmitting the data packets received from the emulated electronic device comprises retrieving the data packets from the second buffer.

3. (Previously presented) The method of Claim 1, wherein the first buffer comprises a receive buffer and a transmit buffer, said method further comprises:

storing the data packets received from the network in the receive buffer; and
transferring the data packets stored in the receive buffer to the transmit buffer.

4. (Original) The method of Claim 2, wherein the second buffer comprises a receive buffer and a transmit buffer, said method further comprises:

storing the data packets received from the network in the receive buffer; and
transferring the data packets stored in the receive buffer to the transmit buffer.

5. (Original) The method of Claim 1, further comprising changing the size of the first buffer at run time.

6. (Original) The method of Claim 1, further comprising discarding packets of data when the first buffer is full.

7. (Original) The method of Claim 1, further comprising keeping a record of the data packets received from the network, the data packets transmitted to the emulated electronic device, the data packets received from the emulated electronic device; and the data

packets transmitted to the network.

8. (Original) The method of Claim 7, further comprising displaying the record on a screen.

9. (Original) The method of Claim 7, further comprising storing the record in a file.

10. (Original) The method of Claim 1, further comprising recording the throughput of the data packets.

11. (Original) The method of Claim 1 further comprising modifying the packets to make the packets suitable for receipt by the emulated device.

12. (Original) The method of Claim 11 wherein modifying includes removing a preamble from a data packet.

13. (Previously presented) The method of Claim 1, wherein the receiving data packets from the network, and the storing the data packets received from the network and the transmitting the data packets in the first buffer are executed in a first thread and the receiving data packets from the emulated electronic device and the transmitting the data packets received from the emulated electronic device are executed in a second thread.

14. (Previously presented) The method of Claim 1, wherein the receiving data packets from the network and the storing of data packets received from the network are executed in a first thread, the transmitting the data packets received in the first buffer is executed in a second thread, the receiving data packets from the emulated electronic device and the transmitting the data packets received from the emulated electronic device are executed in a third thread.

15. (Previously presented) The method of Claim 1, wherein the receiving data packets from the network and the storing of data packets received from the network are executed in a first thread, the transmitting the data packets in the first buffer is executed in a second thread, the receiving data packets from the emulated electronic device is executed in a third thread, and the transmitting the data packets received from the emulated electronic device is executed in a fourth thread.

16. (Previously presented) The method of Claim 2, wherein the receiving data packets from the network and the storing of data packets received from the network are executed in a first thread, the transmitting the data packets in the first buffer is executed in a second thread, the receiving data packets from the emulated electronic device and storing the data packets received from the emulated electronic device are executed in a third thread, and the transmitting the data packets received from the emulated electronic device is executed in a fourth thread.

17. (Previously presented) A method as in Claim 1, further comprising, for testing the operation of the computer executing the network handling program:

generating a data packet in a second computer;

transmitting the data packet, from the ~~first~~ second computer to the computer executing the network handling program;

transmitting back the data packet received by the computer executing the network handling program to the ~~first~~ second computer;

comparing the data packet received by the second computer with the data packet that was sent by the computer executing the network handling program; and

reporting an error if the data packet received by the second computer does not match the data packet that was sent by the computer executing the network handling program.

18. (Previously presented) A method-as in Claim 17, wherein the step of transmitting the data packet comprises:

at the computer executing the network handling program, transmitting the data stored in the first buffer to a third computer;

at the third computer, transmitting back the data packet received to the computer executing the network handling program; and

at the computer executing the network handling program, transmitting the data received from the third computer to the second computer.

19. (Previously presented) An apparatus for connecting an emulated electronic device to a network running at a specified bit-rate higher than the emulated electronic device, the apparatus comprising:

a computer having a memory, a network interface capable of operating at the specified bit-rate for connecting the computer to the network and an emulation interface for connecting the computer to the emulated electronic device; and

computer instructions executable by the computer for:

creating a first buffer in the memory;

receiving data packets from the network through the network interface;

storing data packets received from the network in the first buffer;

transmitting the data packets in the first buffer to the emulated electronic device through the emulation interface;

receiving the data packets from the emulated electronic device through the emulation interface; and

transmitting the data packets received from the emulated electronic device to the network through the network interface.

20. (Previously presented) The apparatus of Claim 19 wherein the emulation interface comprises:

a parallel port card; and

a parallel port cable to connect the computer to the electronic device.

21. (Currently amended) The apparatus of Claim 19, wherein the emulation interface comprises:

a bi-directional interface card; and

a bi-directional interface cable.

22. (Canceled)

23. (Original) The method of Claim 2, further comprising discarding data packets when the second buffer is full.

24. (Original) The method of Claim 3, further comprising discarding data packets

when either one of the receive buffer and the transmit buffer is full.

25. (Original) The method of Claim 4, further comprising discarding data packets when either one of the receive buffer and the transmit buffer is full.